Which Feature is Unusable? Detecting Usability and User Experience Issues from User Reviews

Elsa Bakiu
Technische Universität München
Germany
bakiu@tum.de

Emitza Guzman
University of Zurich
Switzerland
guzman@ifi.uzh.ch

***Abstract*—Usability and user experience (UUX) strongly affect software quality and success. User reviews allow software users to report UUX issues. However, this information can be difficult to access due to the varying quality of the reviews, its large numbers and unstructured nature. In this work we propose an approach to automatically detect the UUX strengths and issues of software features according to user reviews. We use a collocation algorithm for extracting the features, lexical sentiment analysis for uncovering users' satisfaction about a particular feature and machine learning for detecting the specific UUX issues affecting the software application. Additionally, we present two visualizations of the results. An initial evaluation of the approach against human judgement obtained mixed results.**

***Index Terms*—user feedback; software evolution; text mining; user experience; usability.**

I. Introduction

Usability and user experience (UUX) are essential for promoting software success [1], [2]. However, the analysis of the UUX of a product—software included—usually requires expensive evaluations which are often executed in non-natural environments. Furthermore, such evaluations often span for a short period of time, making it difficult to evaluate specific UUX dimensions which require longer term evaluations, such as learnability or memorability.

Previous research has shown that user reviews are a valuable source of UUX information [3]. Users can write these reviews in specialized review sites (e.g., epinions.com), in application distribution platforms (e.g., Google Play, App Store and Amazon) and social media platforms (e.g., Twitter, Facebook and Blogger). These reviews not only contain general software assessments or recommendations, but also include relevant information that can drive the software development effort, such as reports of users' experiences when employing the software in a particular context [1] [4]. A previous study [3] found that 49% of the user review content contains UUX information that could be used to improve the software. However, it is difficult to manually extract this information, due to the large number of reviews (in the order of thousands per day for popular software applications), their lack of structure and varying quality.

In this work, we present an approach for automatically extracting and visualizing users' satisfaction with the UUX of specific software features, as expressed in user reviews. We use a collocation algorithm for automatically extracting the mentioned features, lexical sentiment analysis to extricate the user satisfaction associated to each feature and supervised machine learning for extracting specific UUX information (i.e., information concerning the softwares' memorability, learnability, efficiency, among other aspects). Additionally, we present two visualizations that display the extracted information.

To our best knowledge, our work is the first to combine the aforementioned techniques for extracting UUX information from user reviews on a feature level. We believe that our approach can be useful for aiding developers, UUX designers and researchers to uncover users' satisfaction concerning UUX aspects of specific software features.

II. Definitions

In the context of our work, we use the following *usability*, *user experience* and *feature* definitions:

- **Usability** is the ease of use and acceptance of a product, system or service by users [5]. Its determinants are the concerned product, system or service, the user and the organizational and environmental context [5].
- **User Experience** is the perceptions and responses in the user that result from the use or anticipated use of a product, system or service [6]. User experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use. As with usability, three factors influence user experience: the concerned product, system or service, the user and the context of use [6].
- **Feature** is any prominent or distinctive characteristic or quality of an app [7]. It can be any description of specific software functionality visible to the user, a specific screen of the software application, a general quality of the software, as well as specific technical characteristics.

III. Approach

The main goal of the approach is to automatically extract the level of user satisfaction regarding the UUX of specific software features. Figure 1 shows an overview of our approach. First, we *extract features* mentioned in user reviews by using a collocation algorithm. Then, we apply *sentiment analysis* on the user reviews and map the extracted sentiments to the uncovered features. With this step we detect the satisfaction of the user with the concerned feature. Additionally, we

978-1-5386-3488-2/17 $31.00 © 2017 IEEE
DOI 10.1109/REW.2017.76

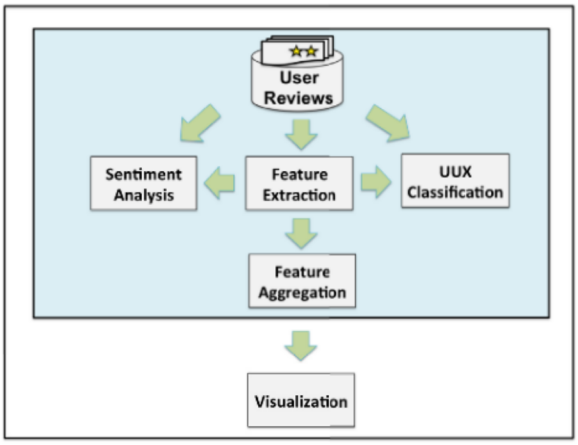

Fig. 1: Overview of the main steps of the approach.

use machine learning classifiers to automatically *classify UUX* issues/strengths (e.g., memorability, likability, efficiency) related to the features mentioned in each review. Afterwards, we *aggregate* all extracted information about the specific features and *visualize* all the mined information. We subsequently detail each of the main steps.

A. Feature Extraction

For the feature extraction, we use the same steps described in previous work [8]. We first filter the review text so that only nouns, verbs and adjectives are taken into account, since these are the parts of speech that generally describe features. For this, we use the POS tagger of NLTK[1]. Furthermore, we remove stopwords and words that are not present in traditional stopword lists, but are common in user reviews and not used to describe software features. These terms are the name of the concerned software, as well as the terms "please", and "fix". Then, we apply the collocation finding algorithm of the NLTK toolkit. A collocation is a set of words that appear together unusually often within a certain distance. For example <*black tea*> is a collocation because it appears unusually often, whereas <*blue tea*> is not because the set of words "blue" and "tea" rarely appear together. Features can usually be described through collocations, as they are usually depicted as a collection of terms that are used together (or withing a certain distance) repeatedly. For example, the feature descriptors <*pdf viewer*>, <*user interface*> and <*view picture*> are sets of words that appear together (or within a certain distance) unusually often. We use a likelihood-ratio test [9] for finding collocations in our reviews.

Afterwards, we remove all detected collocations that only appear in less than three reviews and that have a distance of less than three words between them. While in collocations order is considered important, we ignore word ordering for the purpose of extracting feature descriptors. Therefore, we consider the set of words $< w_i, w_j >$ the same as the set $< w_j, w_i >$. For example, we consider that the pairs of words <*pdf viewer*> and <*viewer pdf*> describe the same feature and therefore group them into one single feature descriptor (the most popular among the set of analyzed reviews). Lastly, we group collocations whose pairs of words are synonyms and use Wordnet[2] as a dictionary. Wordnet also allows the consideration of collocations with misspellings.

B. Sentiment Analysis

Sentiment analysis is the process of assigning a quantitative value (positive or negative) to a piece of text expressing an affect or mood [10]. We extract the sentiment in the user reviews by using SentiStrength [11], a lexical sentiment analysis tool specialized in short, informal text.

SentiStrength divides the input review text into sentences and then assigns a positive and negative value to each one. It assigns positive scores in the [+1,+5] range, where +5 denotes an extremely positive sentiment and 1 denotes the absence of sentiment. Similarly, negative sentiments range from [-5,-1].

Only words that are present in a predefined dictionary are attributed with an individual score. Modifier words (e.g. "really", "very") emoticons and symbols (e.g., "!", "???") also alter the score. The sentiment score of a sentence is calculated by taking the maximum and minimum scores among all the words in the sentence.

We consider the sentiment score of a feature to be equal to the maximum absolute value of the positive and negative score of the sentence in which it is present. For the cases where the positive and negative values are the same, we assign the negative value to the feature. For example, lets consider the sentence "Uploading pictures with the app is so terrible!". This sentence contains the feature <*upload picture*>, and the sentiment score of the sentence is 1,-4. Therefore, we assign the feature the sentiment score of -4. This step produces a list of all extracted features, their frequencies (how often they were mentioned), and their sentiment scores.

C. UUX Classification

The goal of this step is to automatically detect specific UUX information associated to each extracted feature. We use the UUX dimensions presented in previous work [3] as our taxonomy for characterizing UUX. This taxonomy unifies the five dimensions of usability defined by Nielsen [12] and the set of dimensions from three popular studies of user experience [13], [14], [15]. Table I summarizes the UUX dimensions considered in this work.

We use supervised machine learning techniques to classify the described issues/strengths of extracted features into the corresponding UUX dimensions. Each feature can be associated to more than one UUX dimension, i.e., a feature can be associated with a memorability and efficiency issue.

In machine learning, the classification of documents (description of issues/strengths faced by features in our case) into more than one label (UUX dimensions in our approach)

[1]http:www.nltk.org

[2]https:wordnet.princeton.edu

Nielsen	Bevan	Ketola	Bargas-Avila
Memorability	Likeability	Anticipation	Affect and Emotion
Learnability	Pleasure	Overall Usability	Enjoyment and Fun
Efficiency	Comfort	Hedonic	Aesthetics and Appeal
Errors/Effectiveness	Trust	Detailed usability	Engagement
Satisfaction		User Differences	Motivation
		Support	Enchantment
		Impact	Frustration
			Hedonic

Table I: Dimensions of UUX used in this study.

is referred to as multi-label classification. We use the most popular multi-labeling solution, the binary relevance method (BR). BR consists of decomposing the multi-label problem into several independent binary classification problems, one for each label. The final multi-label prediction for a new instance is determined by aggregating the classification results from all independent binary classifiers. Additionally, we use SVM as the classifier to perform the classification task, due to its good performance when classifying text [16].

We trained our classifier on a manually labeled collection of software reviews [3] from two product categories: software (520 reviews) and video games (2972 reviews). This dataset contains manually assigned labels for the UUX dimensions presented in Table I on the sentence level. We argue that this granularity is enough for the extraction of UUX dimensions on the feature level as it is often the case that no more than one feature is mentioned in the sentence of a user review.

We train our classifier by inputting all of the sentences including at least one feature according to the feature extraction step (see Section III-A).

Before inputting the data to the classifier, we perform the following preprocessing steps: (1) identify and remove stop-words using the English language stop-words list provided in the NLTK library, (2) stem the text using the NLTK Snowball stemming algorithm to reduce words to their grammatical roots so that they can be represented by a single term, (3) transform the text into a bag of words representation using TFxIDF as a weighting factor and finally, (4) select the best features according to the Chi-Squared metric.

Afterwards, we use the trained classifier to make predictions about the UUX dimensions of unseen sentences of user reviews. The classifier then assigns a UUX dimension(s) to each input sentence. We then map the predicted UUX dimension(s) to the feature(s) contained in the analyzed sentence. A list of the extracted features and its assigned UUX dimensions is the output of this step.

We use the SVM and BR implementation of scikit-learn[3], a machine learning library for Python.

D. Feature Level Aggregation

We aggregate the extracted information for each feature by averaging the sentiment and unifying the UUX dimensions of all extracted features sharing the same name. For example if the feature *<pdf viewer>* appears three times in the analyzed reviews: (1) with sentiment score 2 and UUX dimensions *learnability* and *efficiency*, (2) with sentiment score 3 and UUX dimension *support* and (3) with sentiment score 1 and dimension *learnability*, then the aggregated feature *<a—pdf viewer>* will have an aggregated sentiment score of 2 and will be associated to the UUX dimensions *learnability*, *efficiency* and *support*. The output of this step is a list of features with its aggregated sentiment and UUX dimensions.

[3]http:www.scikit-learn.org/

E. Visualization

In this step we visualize the mined information. Visualization can aid human cognition by leveraging the visual capacity for identifying patterns, trends and outliers, making it easier for developers and analysts to interpret the mined data. For our visualizations, we apply the Information Seeking Mantra proposed by Schneiderman [17]: overview first, then zoom and filter, details on demand. Thus, we visualize the results of the feature level aggregation in two granularity levels: high-level and detailed.

Figure 2 provides the high-level visualization, which displays a general overview of the sentiment of UUX dimensions across the most popular features, the rating of the user reviews over time and the amount of reviews with a positive, negative and neutral sentiment. Users can then use the more detailed visualization, shown in Figure 3 for getting additional information about the user acceptance of specific UUX dimensions concerning specific features. For both visualizations users can see examples of the actual review text associated to the visualization on demand by placing the cursor on specific points.

The detailed UUX visualization could be especially valuable to the development team by helping them identify the most and least popular features and take decisions on how to improve their UUX aspects during software evolution.

IV. Preliminary Evaluation

We assessed the feasibility of our approach by performing a preliminary evaluation. First, we ran our approach on a collection of user reviews about video games and other software gathered in previous work [3]. Then, we performed an opportunistic sampling of 12 correctly extracted and often mentioned features[4]. We only considered correctly extracted features, as we evaluated the quality of the results of the feature extraction step in our previous work [8]. Afterwards, we compared the results produced by our approach in the sentiment analysis and UUX classification steps against a golden standard.

For the creation of the golden standard, we manually assigned these features a sentiment on a 5 level scale (from "very positive" to "very negative") and also manually annotated the UUX dimensions associated to them. We reduced subjective assessments by using the description of each dimension provided in previous work [3] as an annotation guideline for the UUX dimensions. For the sentiments we used a guideline we created in our previous work [8]. We analyzed each of the individual sentences that contained each the selected 12

[4]the data that was used in the evaluation was excluded when training the classifier described in Section III-C

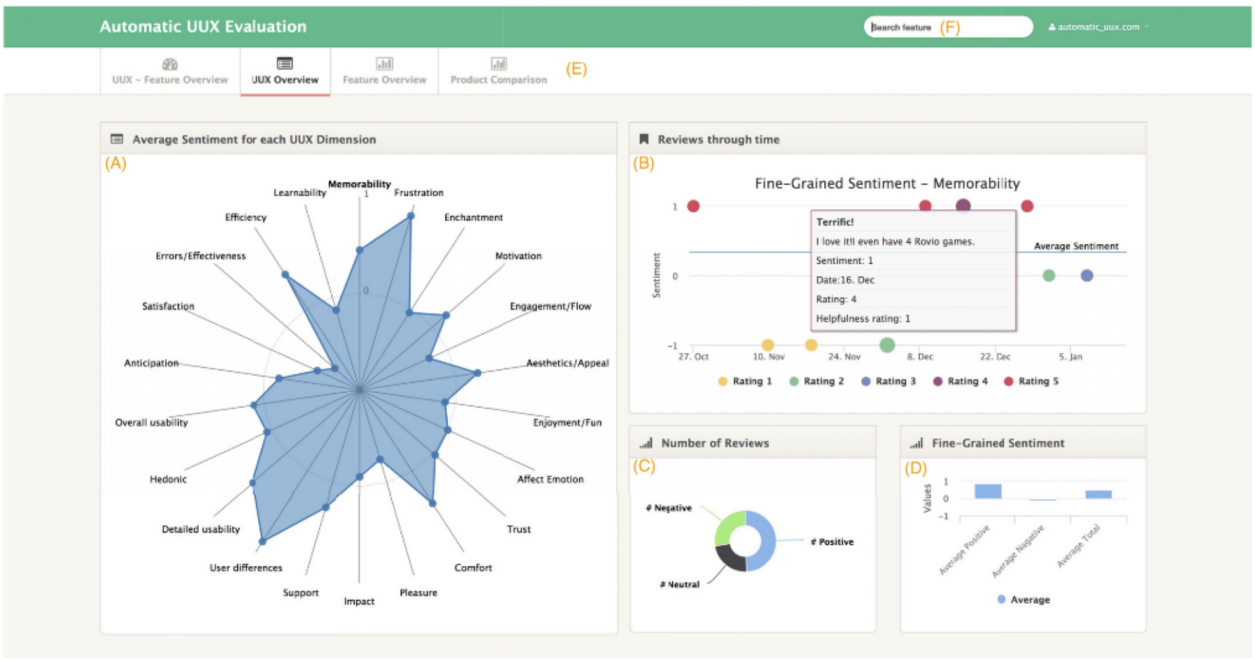

Fig. 2: Overview UUX Visualization. The following information is depicted: (A) Number of relevant reviews and average sentiment of each UUX dimension about the software in general (or about the features the user has searched for), (B) Reviews pertaining to the selected UUX dimension over time, (C) Number of positive, negative and neutral reviews pertaining to the UUX dimension, (D) Average positive sentiment, average negative sentiment and overall average sentiment of the selected UUX dimension, (E) Navigation menu, (F) Search field to query features.

Dimension & Frequency	Precision	Recall	F-measure	Dimension & Frequency	Precision	Recall	F-measure
Memorability (2)	1.0	0.50	0.67	Pleasure (6)	0.5	0.25	0.33
Learnability (5)	1.0	0.4	0.57	Trust (1)	0	0	0
Errors/Effectiveness (5)	1.0	0.4	0.57	Affect and Emotion (6)	0.5	0.33	0.40
Satisfaction (31)	0.81	0.68	0.73	Enjoyment, Fun (6)	0.5	0.5	0.5
Engagement and Flow (16)	0.9	0.53	0.67	Aesthetics and Appeal (13)	0.92	0.92	0.92
Detailed Usability (37)	0.68	0.79	0.73	Enchantment (1)	0	0	0
Hedonic (7)	0.8	0.5	0.62	Frustration (2)	1.0	0.5	0.67

Table II: UUX Classification results at the Feature level. Under parenthesis the frequency of appearance of each dimension in the manually annotated test set. Dimensions without instances are not reported.

features and assigned individual sentiments and dimensions for every feature appearance. This resulted in a total of 70 analyzed sentences—each associated to the individual features.

We use the standard precision, recall and F-Measure to report on our results. Table II shows an overview of our results for the feature UUX classification. The results of this step were mixed among the dimensions. In general, the dimensions with a larger presence in the test set (present more than 15 times) achieved better results than those that were more poorly represented. A possible reason for this, could also be their dominance in the training set used by the classifier. The sentiment analysis achieved a precision of 0.68, recall of 0.64 and F-measure of 0.64. These results could be improved by extending the dictionary of the tool used in the sentiment analysis step to include software engineering and UUX concepts, or by applying machine learning techniques.

V. Discussion

The results of our preliminary evaluation are mixed and a more extensive evaluation is needed to have more conclusive results about the effectiveness of the approach. Future evaluations should be conducted on all individual steps of the approach against a larger test set. Different approaches for feature extraction, sentiment analysis and UUX classification could then be compared. Additionally, further visualizations could be designed and their impact and cognitive overload could be evaluated.

We must note that the potential usefulness of user reviews has some important caveats when analyzing UUX characteristics. The reviews contain very few details about reviewers (e.g., gender, age, preferences), in contrast to standard UUX studies. In addition, some of the reviews might be fake and are only written after the launching of the product to the market.

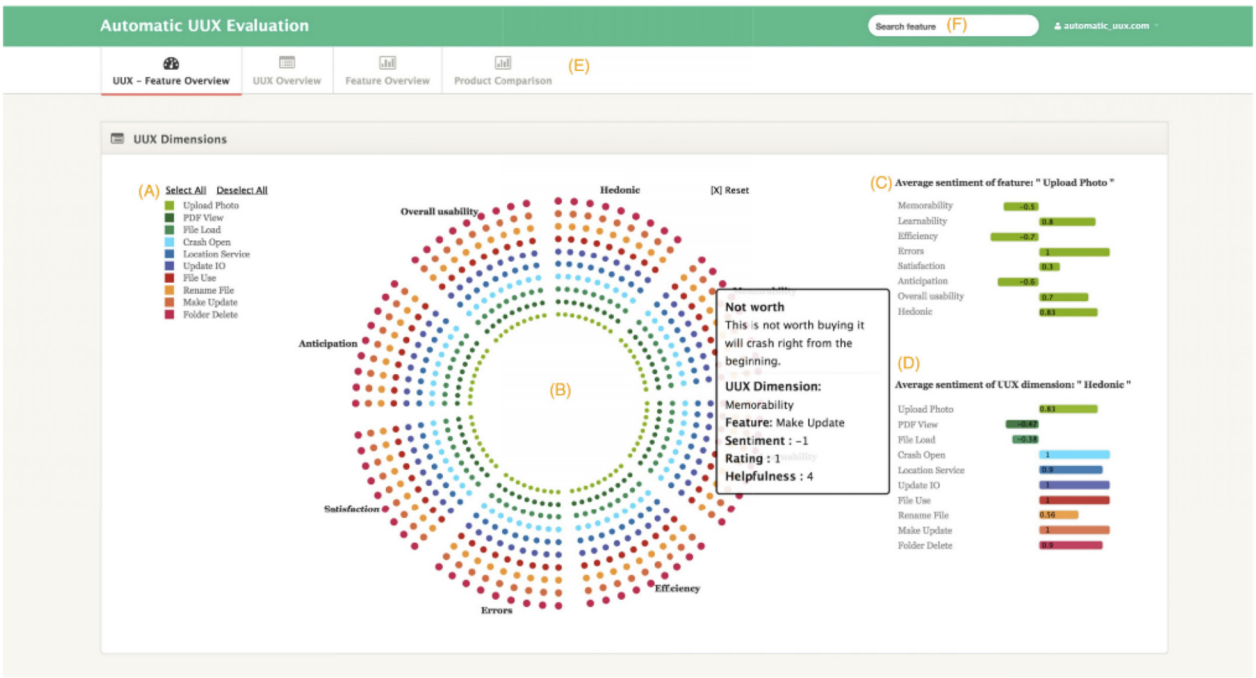

Fig. 3: Detailed Feature-UUX Visualization. The following information is depicted: (A) Most popular features (or the features the user has searched for) concerning specific UUX dimensions, (B) Examples of some of reviews for each feature - UUX dimension pair, (C) Average sentiment of the selected feature for each UUX dimension, (D) Average sentiment of the selected UUX dimension for each feature, (E) Navigation menu, (F) Search field to query features.

Furthermore, Hu et al. [18] found that very satisfied or very dissatisfied customers are more likely to engage in writing reviews, which leads us to infer that in terms of satisfaction, the average user is underrepresented among reviewers, and that reviews may not always yield a representative description of the typical experiences among the user base. Therefore, the evaluation of UUX on software features by analyzing the user reviews cannot be a replacement for the already existing methods, but rather an addition to address their limitations and to provide useful information for software evolution.

VI. Related Work

A. User Feedback and Crowd-based Requirements Engineering

Researchers have explored user involvement in requirements and software engineering [19] and have coined the term crowd-based requirements engineering [20] for describing the contribution of users to different requirements engineering activities. Work in this direction is driven by the rise of social media and mobile applications.

Seyff et al. [21] suggested to use existing social network sites and explored the use of Facebook for requirements elicitation, prioritization and negotiation. Pagano and Maalej [22] and Hoon [23] described exploratory studies in which they analyzed the amount, content and rating characteristics of user feedback from mobile application distribution platforms (i.e., app stores), which allow for the elicitation of requirements from distributed users. Seyff et al. [24] and Schneider et al. [25] presented approaches for continuously eliciting feedback from mobile devices. Pagano et al. [26] discussed how user feedback can be considered in software development in general.

We believe that the approach presented in this work will aid in the crowdsourcing of requirements concerning UUX aspects.

B. Mining User Feedback

User feedback mining has received sizable attention in recent past years. We believe that our work is the first to propose to mine UUX issues/strengths on a feature level.

However, previous work has previously extracted features mentioned in user feedback [8],[27],[28], extracted the sentiment from the reviews using lexical sentiment analysis [8] and machine learning [29], classified user feedback into the UUX dimensions used in this work [3] and visualized user feedback [30]. Machine learning approaches have often been applied for automatically classifying user feedback into categories relevant for software evolution e.g., [4], [31], [29]. Existing work has also focused on summarizing user feedback by applying topic modeling [32], [33], as well as on the prioritization of the feedback [34], [35], [36], and the retrieval of diverse feedback [37].

The main difference of this work to previous one is its proposal to use existing techniques on a feature level granularity to aid developers, UUX experts (practitioners and researchers)

and end-users to identify UUX issues/strengths of software features mentioned in user reviews.

VII. Conclusions

We described an approach to extract users' satisfaction about the UUX of specific software feature. We believe that our approach can aid developers, as well as UUX designers and researchers to detect concrete points of improvement concerning the UUX of current software applications.

References

[1] T. Chattopadhyay and N. Natrajan, "The role of functional requirements, operational quality and usability in the success story of software projects," *International Journal of Engineering Science Invention*, pp. 36–41, 2013.

[2] F. Lizano, M. M. Sandoval, A. Bruun, and J. Stage, "Is usability evaluation important: The perspective of novice software developers," in *Proceedings of the 27th International BCS Human Computer Interaction Conference*. British Computer Society, 2013, p. 31.

[3] S. Hedegaard and J. G. Simonsen, "Extracting usability and user experience information from online user reviews," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 2089–2098.

[4] E. Guzman, M. El-Haliby, and B. Bruegge, "Ensemble Methods for App Review Classification: An Approach for Software Evolution (N)," in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE'15, 2015, pp. 771–776.

[5] N. Bevan, "What is usability," in *In: Human Aspects in Computing: Design and Use of Interactive Systems with Terminals*, 1991, pp. 651–655.

[6] ISO, "Iso 9241-210: Ergonomics of human system interaction - part 210: Human-centered design for interactive systems." International Organization for Standardization, Standard, 2009.

[7] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," DTIC Document, Tech. Rep., 1990.

[8] E. Guzman and W. Maalej, "How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews," in *Proceedings of the 2014 IEEE 22nd International Requirements Engineering Conference*, ser. RE'14, 2014, pp. 153–162.

[9] C. D. Manning, H. Schütze *et al.*, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.

[10] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A Large-scale Sentiment Analysis for Yahoo! Answers," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12, 2012, pp. 633–642.

[11] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment Strength Detection in Short Informal Text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[12] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[13] N. Bevan, "Classifying and selecting UX and usability measures," in *International Workshop on Meaningful Measures: Valid Useful User Experience Measurement*, 2008, pp. 13–18.

[14] P. Ketola, in *Proceedings of the Open Workshop on Valid Useful User Experience Measurement*, ser. VUUM '08.

[15] J. A. Bargas-Avila and K. Hornbæk, "Old wine in new bottles or novel challenges: A critical analysis of empirical studies of user experience," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2689–2698.

[16] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[17] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings of the IEEE Symposium on Visual Languages*. IEEE, 1996, pp. 336–343.

[18] N. Hu, P. A. Pavlou, and J. Zhang, "Can online reviews reveal a product's true quality?: Empirical findings and analytical modeling of online word-of-mouth communication," in *Proceedings of the 7th ACM Conference on Electronic Commerce (EC 2006)*, 2006, pp. 324–330.

[19] T. Johann and W. Maalej, "Democratic Mass Participation of Users in Requirements Engineering?" in *Proceedings of the 2015 IEEE 23rd International Requirements Engineering Conference*, ser. RE'15, 2015, pp. 256–261.

[20] E. C. Groen, J. Doerr, and S. Adam, "Towards Crowd-Based Requirements Engineering A Research Preview," in *Requirements Engineering: Foundation for Software Quality*. Springer, 2015, pp. 247–253.

[21] N. Seyff, I. Todoran, K. Caluser, L. Singer, and M. Glinz, "Using Popular Social Network Sites to Support Requirements Elicitation, Prioritization and Negotiation," *Journal of Internet Services and Applications*, vol. 6, no. 1, pp. 1–16, 2015.

[22] D. Pagano and W. Maalej, "User Feedback in The Appstore: an Empirical Study," in *Proceedings 2013 21st IEEE International Requirements Engineering Conference*, ser. RE'13, 2013, pp. 125 – 134.

[23] L. Hoon, R. Vasa, J.-G. Schneider, J. Grundy, and Others, "An Analysis of The Mobile App Review Landscape: Trends and Implications," *Faculty of Information and Communication Technologies, Swinburne University of Technology, Tech. Rep*, 2013.

[24] N. Seyff, G. Ollmann, and M. Bortenschlager, "AppEcho: A User-driven, in Situ Feedback Approach for Mobile Platforms and Applications," in *Proceedings of the 1st International Conference on Mobile Software Engineering and Systems*, ser. MOBILESoft'14. ACM, 2014, pp. 99–108.

[25] K. Schneider, S. Meyer, M. Peters, F. Schliephacke, J. Mörschbach, and L. Aguirre, "Feedback in context: Supporting the evolution of it-ecosystems," in *International conference on product focused software process improvement*. Springer, 2010, pp. 191–205.

[26] W. Maalej and D. Pagano, "On the socialness of software," in *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC),*. IEEE, 2011, pp. 864–871.

[27] X. Gu and S. Kim, ""What Parts of Your Apps are Loved by Users?" (T)," in *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE'15, 2015, pp. 760–770.

[28] P. M. Vu, T. T. Nguyen, H. V. Pham, and T. T. Nguyen, "Mining user opinions in mobile app reviews: A keyword-based approach (t)," in *Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on*. IEEE, 2015, pp. 749–759.

[29] S. Panichella, A. Di Sorbo, E. Guzman, C. Visaggio, G. Canfora, and H. Gall, "How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution," in *Proceedings of the 31st International Conference on Software Maintenance and Evolution*, ser. ICSME'15, 2015, pp. 281 – 290.

[30] E. Guzman, P. Bhuvanagiri, and B. Bruegge, "Fave: Visualizing user feedback for software evolution," in *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. IEEE, 2014, pp. 167–171.

[31] W. Maalej and H. Nabil, "Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews," in *Proceedings of the 23rd International Requirements Engineering Conference*, ser. RE'15, 2015, pp. 116–125.

[32] L. V. Galvis Carreño and K. Winbladh, "Analysis of User Comments: An Approach for Software Requirements Evolution," in *Proceedings of the 2013 International Conference on Software Engineering*, ser. ICSE'13, 2013, pp. 582–591.

[33] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in *Proceddings of the Working Conference on Mining Software Repositories*, ser. MSR'13, may 2013, pp. 41–44.

[34] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace," in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE'14, 2014, pp. 767–778.

[35] L. Villarroel, G. Bavota, B. Russo, R. Oliveto, and M. Di Penta, "Release Planning of Mobile Apps Based on User Reviews," in *Proceedings of the 38th International Conference on Software Engineering*, ser. ICSE'16. ACM, 2016, pp. 14–24.

[36] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora, and H. C. Gall, "What Would Users Change in My App? Summarizing App Reviews for Recommending Software Changes," in *Proceedings of the International Symposium on Foundations of Software Engineering*, 2016, pp. 499–510.

[37] E. Guzman, O. Aly, and B. Bruegge, "Retrieving Diverse Opinions from App Reviews," in *Proceedings of the 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM'15, 2015, pp. 1–10.